

INTERIM LOW BIT-RATE IMAGE COMPRESSION

**FOR THE
NATIONAL IMAGERY TRANSMISSION FORMAT
STANDARD**

DRAFT

FOREWORD

1. The National Imagery Transmission Format Standard (NITFS) is the standard for formatting digital imagery and imagery-related products and exchanging them between members of the Intelligence Community (IC) as defined by Executive Order 12333, the Department of Defense (DoD), and other departments or agencies of the United States Government as governed by Memoranda of Agreement with those departments or agencies.

2. The National Imagery Transmission Format Standard Technical Board (NTB) developed this standard based upon currently available technical information.

3. The DoD and other IC members are committed to interoperability of systems used for formatting, transmitting, receiving, and processing imagery and imagery-related information. This standard describes the interim low bit-rate compression algorithm and establishes its application within the NITFS.

4. Beneficial comments (recommendations, additions, deletions) and other pertinent data which may be of use in improving this document, should be addressed to: Defense Information Systems Agency (DISA), Joint Interoperability and Engineering Organization (JIEO), Center for Standards (CFS), Attention: TBCE, 10701 Parkridge Boulevard, Reston, VA 22091-4398 by using the pre-addressed Standardization Document Improvement Proposal (DD Form 1426) appearing at the end of this document, or by letter.

Contents

1. SCOPE.....	1
1.1 Scope	1
1.2 Content.....	1
1.3 Applicability	1
1.4 Tailoring of task, method, or requirement specifications	1
1.5 Types of operation	1
2. APPLICABLE DOCUMENTS	3
2.1 Government documents	3
2.1.1 Specifications, standards, and handbooks	3
2.1.2 Other Government documents, drawings, and publications	3
2.2 Non-Government publications	4
2.3 Order of precedence	4
3. DEFINITIONS, ACRONYMS AND SYMBOLS	5
3.1 Acronyms used in this standard	5
3.2 Definitions used in this standard	5
3.3 Symbols used in this standard	5
4. GENERAL REQUIREMENTS	7
4.1 Interoperability	7
4.2 Encoders	7
4.2.1 Image downsampling	7
4.2.2 JPEG encoding	7
4.3 Decoders	8
4.3.1 JPEG decoding	8
4.3.2 Image upsampling	8
4.4 Interchange format-encoders.....	9
4.5 Interchange format-decoders.....	9
4.6 Further general requirements	10
5. DETAILED REQUIREMENTS	10
5.1 Image downsampling process	11
5.1.1 Downsampled image dimensions	11
5.1.2 Downsampling filter operation	12
5.1.2.1 Downsample filter parameter calculations	13
5.1.3 Application of the downsampling filter	14
5.1.3.1 Downsampling along the image edges	14
5.2 JPEG compression of the downsampled image	16
5.2.1 Control procedures for the sequential DCT mode	16
5.2.2 Compressed data interchange format	16
5.2.2.1 Format of a JPEG compressed image within an NITF file	16
5.2.2.1.1 Single block JPEG compressed format	16
5.2.2.1.1.1 Single block image data format	17
5.2.2.1.1.2 Frame and Scan formats	17
5.2.2.1.2 Multiple block JPEG compressed format	17

5.2.2.1.2.1 Multiple block image data format	18
5.2.2.1.2.2 Frame and Scan formats	18
5.2.2.2 NITF image subheader	18
5.2.2.3 APP ₆ “NITF” application data segment	19
5.3 JPEG decompression of the downsampled image	20
5.4 Image upsampling process	21
5.4.1 Upsample ratio calculation	21
5.4.2 Upsampling filter operation	21
5.4.2.1 Upsample filter parameter calculations	22
5.4.3 Application of the upsampling filter	23
5.4.3.1 Upsampling along the image edges	23
6. NOTES	26
6.1 Downsample ratio calculation	26
6.2 Image upsampling	26
6.3 Overlays	27
6.4 Inherent quality losses	27
6.5 Incompatibility issues with previous generations of NITFS systems	27
APPENDIX A	28
A.1 Recommended compression parameters	28
A.2 Eight-bit gray scale compression parameters	28
A.2.1 Downsample ratios	28
A.2.2 JPEG quantization table	28
A.2.3 JPEG Huffman tables	29
A.2.3.1 <u>IQ1</u> Huffman table parameters	29
A.2.3.2 IQ2 Huffman table parameters	30
A.2.3.3 IQ3 Huffman table parameters	30
A.2.3.4 IQ4 Huffman table parameters	32
A.2.3.5 IQ5 Huffman table parameters	32
A.3 Twelve-bit gray scale compression parameters	34
A.3.1 Downsample ratios	34
A.3.2 JPEG quantization table	34
A.3.3 JPEG Huffman tables	34
A.3.3.1 IQ1 Huffman table generation parameters	35
A.3.3.2 IQ2 Huffman table generation parameters	35
A.3.3.3 IQ3 Huffman table generation parameters	36
A.3.3.4 IQ4 Huffman table generation parameters	36
A.3.3.5 IQ5 Huffman table generation parameters	37

Figures

FIGURE 1. NITF file structure	7
FIGURE 2. Interim Algorithm Encoder	8
FIGURE 3. Interim Algorithm Decoder	9
FIGURE 4. Downsampling demonstration with $i=10$, $b_{10}=11$, $e_{10}=15$, and $R = 1.3$	12
FIGURE 5. Illustration of mirroring for image edges	14
FIGURE 6. Control procedure for image downsampling (row-column order)	15
FIGURE 7. NITF single block file structure (IMODE=B or P)	17
FIGURE 8. NITF multiple block file structure (IMODE=B or P)	18
FIGURE 9. Upsampling demonstration with $i=10$, $b_{10}=6$, $e_{10}=9$, and $R = 1.3$	22
FIGURE 10. Control procedure for image upsampling (row-column order)	25

Tables

TABLE I. APP ₆ “NITF” application data segment	20
TABLE A1. Downsample ratios for eight-bit gray scale images	28
TABLE A2. Eight-bit gray scale JPEG quantization table	29
TABLE A3. Downsample ratios for twelve-bit gray scale images	34
TABLE A4. Twelve-bit gray scale JPEG quantization table	34

1. SCOPE

1.1 Scope. This document establishes the requirements to be met by NITFS compliant systems when image data is compressed using the interim low bit-rate image compression algorithm. This algorithm is based on the JPEG DCT image compression algorithm as described in IS 10918-1, *Digital Compression and Coding of Continuous-tone Still Images* and the NITFS implementation of JPEG described in MIL-STD-188-198A, *Joint Photographic Experts Group (JPEG) Image Compression for the National Imagery Transmission Format Standard*.

1.2 Content. This standard provides technical detail of the NITFS compression algorithm designated by the code I1 in the Image Compression (IC) field of the NITF file image subheader for 8-bit and 12-bit gray scale imagery.

1.3 Applicability. This standard is applicable to the Intelligence Community and the Department of Defense. It is mandatory for all Secondary Imagery Dissemination Systems in accordance with the memorandum by the Assistant Secretary of Defense for C³I, Subject: National Imagery Transmission Format Standard (NITFS), 12 August 1991. This directive shall be implemented in accordance with the Joint Interoperability and Engineering Organization (JIEO) Circular 9008, NITFS Certification Test and Evaluation Program Plan, and the MIL-HDBK-1300A. New equipment and systems, those undergoing major modification, or those capable of rehabilitation shall conform to this standard.

1.4 Tailoring of task, method, or requirement specifications. The minimum compliance requirements for implementation of this compression algorithm are defined in JIEO Circular 9008.

1.5 Types of operation. This standard establishes the requirements for the communication or storage for interchange of image data in compressed form. Each type of operation defined by this standard consists of three parts:

- a. The compressed data interchange format (which defines the image data field of the NITF file format).
- b. The encoder.
- c. The decoder.

This standard defines two types of operation:

- | | | |
|-----------|---|--|
| a. Type 1 | - | 8-bit sample precision gray scale sequential Discrete |
| Cosine | | Transform (DCT) with Huffman coding. |
| b. Type 2 | - | 12-bit sample precision gray scale sequential Discrete |
| Cosine | | Transform (DCT) with Huffman coding. |

2. APPLICABLE DOCUMENTS

2.1 Government documents.

2.1.1 Specifications, standards, and handbooks. The following standards form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the issue of the Department of Defense Index of Specifications and Standards (DODISS) and supplements thereto, cited in the solicitation.

FEDERAL STANDARDS

FED-STD-1037B - Telecommunications: Glossary of
Telecommunication Terms, 3 June 1991.

(Copies of the referenced Federal Standards are available from General Services Administration, GSA Specification Section, Room 6654, 7th and D Streets, S.W. Washington, D.C. 20407; telephone (202) 472-2205).

MILITARY STANDARDS

MIL-STD-2500A - National Imagery Transmission Format (Version
2.0) for the National Imagery Transmission
Format Standard, 18 June 1993.

MIL-STD-188-198A - Joint Photographic Experts Group (JPEG) Image
Compression for the National Imagery
Transmission Format Standard.

MILITARY HANDBOOKS

MIL-HDBK-1300A - Military Handbook National Imagery
Transmission Format Standard, 18 June 1993.

(Unless otherwise indicated, copies of federal and military specifications, standards, and handbooks are available from the Standardization Documents Order Desk, 700 Robbins Avenue, Building #4, Section D, Philadelphia, PA 19111-5094).

2.1.2 Other Government documents, drawings, and publications. The following other Government documents form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those cited in the solicitation.

DISA/JIEO Circular 9008 - National Imagery Transmission Format Standard
Certification Test and Evaluation Program Plan,
30 June 1993.

2.2 Non-Government publications. The following documents form a part of this document to the extent specified herein. Unless otherwise specified, the issues of the documents which are Department of Defense (DoD) adopted are those listed in the issue of the DODISS cited in the solicitation.

INTERNATIONAL STANDARDS

ISO IS 10918-1/ CCITT Recommendation T.81	-	Digital Compression and Coding of Continuous-tone Still Images. Part I: Requirements and Guidelines, September, 1992.
--	---	---

ISO IS 10918-3/ CCITT Recommendation T.84	-	Digital Compression and Coding of Continuous-tone Still Images: Extensions, November, 1995.
--	---	---

(Copies may be obtained from X3 Secretariat, Computer and Business Equipment Manufacturers Association, 311 First Street NW, Suite 500, Washington, DC 20001-2178)

NATIONAL STANDARDS

None.

2.3 Order of precedence. In the event of a conflict between the text of this standard and the references cited herein, the text of this standard shall take precedence. Nothing in this standard, however, shall supersede applicable laws and regulations unless a specific exemption has been obtained.

3. DEFINITIONS, ACRONYMS AND SYMBOLS

The following definitions are applicable for the purpose of this standard. In addition, terms used in this standard and defined in the FED-STD-1037B shall use the FED-STD-1037B definition unless noted.

3.1 Acronyms used in this standard.

- a. DCT Discrete Cosine Transform.
- b. JPEG Joint Photographic Experts Group.

3.2 Definitions used in this standard.

- a. Bit-rate. The average number of bits spent per image sample in a compressed image data file.
- b. Downsampling. A process by which an image's dimensions, either the number of samples per row or the number of rows or both, are reduced.
- c. Profile. A specific set of capabilities and parameter values or ranges.
- d. Upsampling. A process by which an image's dimensions, either the number of samples per row or the number of rows or both, are increased.

3.3 Symbols used in this standard.

- a. α Down/upsampling filter length. In the same image space as x_j .
- b. b_i Beginning summation index. In the same image space as x_j .
- c. $\text{ceil}(\cdot)$ Ceiling function, rounds to next integer toward positive infinity.
- d. e_i Ending summation index. In the same image space as x_j .
- e. $\text{floor}(\cdot)$ Floor function, rounds to next integer toward negative infinity.
- f. M_d Number of samples per row in the downsampled image data.
- g. M_o Number of samples per row in the original image data.
- h. N_d Number of rows in the downsampled image data.

- i. N_O Number of rows in the original image data.
- j. R, R_{row}, R_{col} Downsampling ratios. Equal to the ratio of the number of samples in a given dimension of the original image to that of the same dimension in the downsampled image.
- k. $\text{round}(\cdot)$ Round function, rounds toward nearest integer.
- l. w_{ij} Downsampling or upsampling filter coefficient relating x_j to y_i .
- m. x_j Input image sample in the downsampling or upsampling formulae.
- n. y_i Output image sample in the downsampling or upsampling formulae.

4. GENERAL REQUIREMENTS

4.1 Interoperability. The profile specified in this document is intended to enable the interchange in the NITFS format, of 8-bit (Type 1) and 12-bit (Type 2) gray scale imagery compressed with the interim low bit-rate algorithm (also referred to as the interim algorithm) .

4.2 Encoders. Encoders shall output to the image data field of the NITF file a full interchange format that includes the compressed image data and all table specifications used in the encoding process as illustrated in Figure 1 .

4.2.1 Image downsampling. The interim algorithm encoder utilizes a downsampling procedure to extend the low bit-rate performance of the NITFS JPEG algorithm described in MIL-STD-188-198A. Figure 2 illustrates the concept. The downsampling preprocessor allows the JPEG encoder to operate at a higher bit-rate on a smaller version of the original image while maintaining an overall bit-rate that is low.

4.2.2 JPEG encoding. Once downsampling of the original image is performed, the encoding process is identical to that of NITFS JPEG lossy compression algorithm. Minor variations exist in the compressed data format as described in this document. The NITFS JPEG algorithm is a profile of the lossy DCT-based encoding algorithm found in IS 10918-1. Encoders conforming to this standard may use any procedure in IS 10918-1 applicable to DCT encoding subject to the requirements and restrictions expressed in MIL-STD-188-198A and herein.

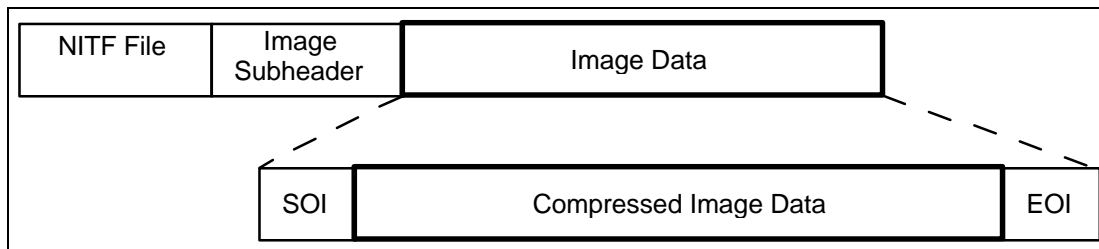
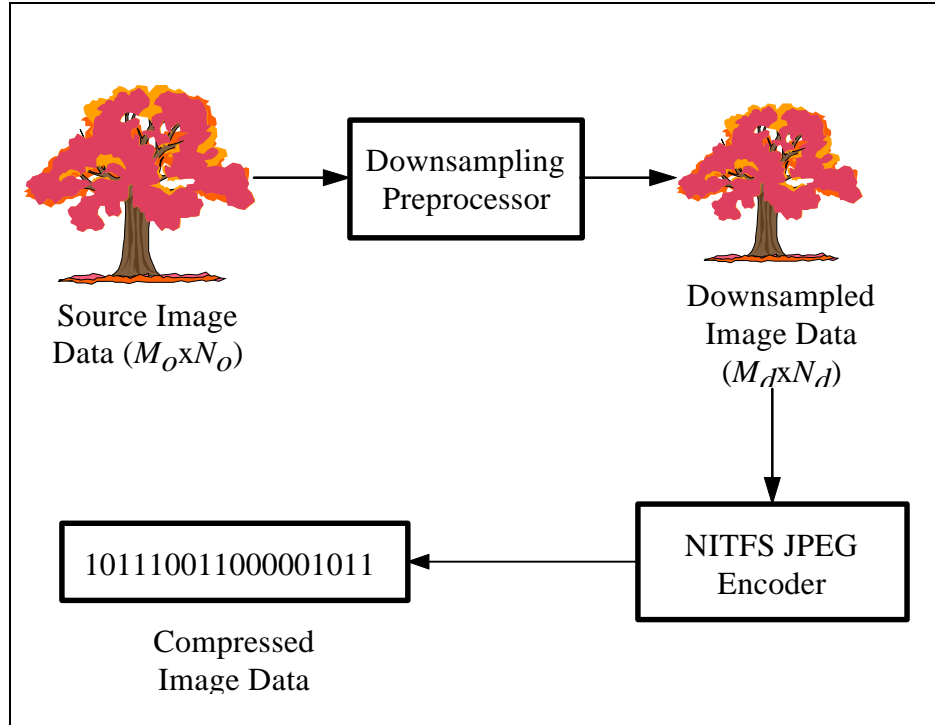


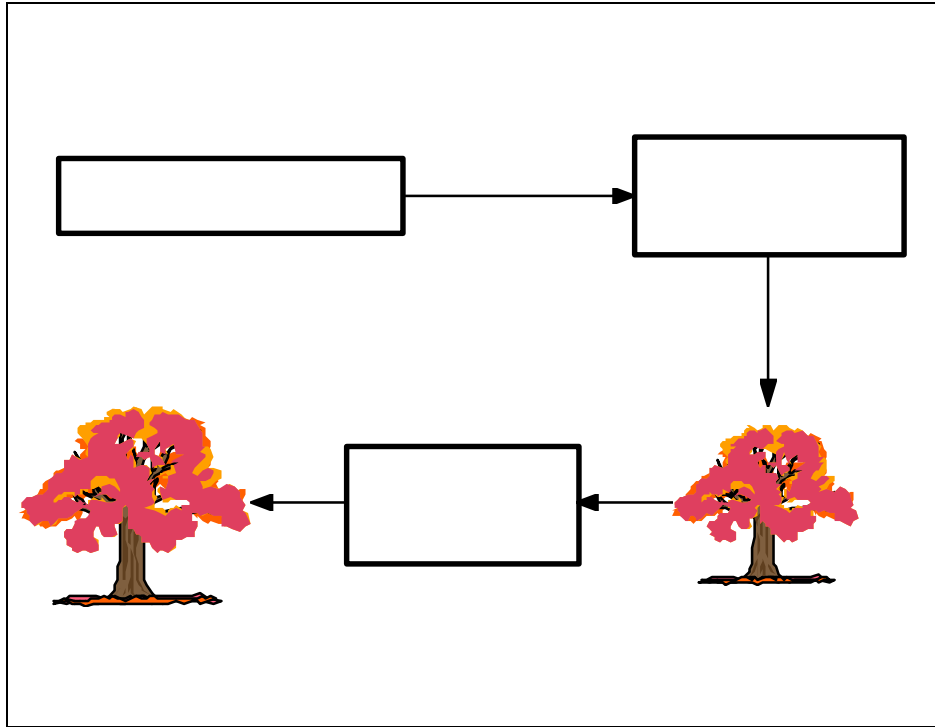
FIGURE 1. NITF file structure.

FIGURE 2. Interim Algorithm Encoder.

4.3 Decoders. All decoders shall interpret full interchange format. Abbreviated interchange format decoders are not a requirement of this profile.

4.3.1 JPEG decoding. The interim algorithm decoder decodes the compressed image data using the NITFS JPEG decoder (see Figure 3). This results in reconstructed image data whose dimensions match that of the downsampled image data in Figure 2.

4.3.2 Image upsampling. An upsampling postprocessor is used to return the reconstructed image data to the same dimensions as that of the original. It is important to note that the down/upsampling processes are not lossless. The interim algorithm makes a tradeoff between JPEG and down/upsampling artifacts in the reconstructed imagery.

FIGURE 3. Interim Algorithm Decoder.

4.4 Interchange format-encoders. Encoders shall output to the image data field of the NITF file either a full or abbreviated interchange format. The full interchange format includes the compressed image data and all table specifications used in the encoding process.

In MIL-STD-188-198A, an abbreviated interchange format is defined that is identical to the full interchange format, except that it does not contain all tables required for decoding a compressed image data file. This capability allows for smaller files, but requires continual maintenance and dissemination of default tables which can not be reliably achieved. Currently default tables are permitted by this standard. However, future NITF compliant systems will be required to embed all necessary tables in the compressed data stream. Implementors are strongly encouraged to avoid usage of default tables.

The tables given in Appendix A of this document are recommended tables. If used, they must be embedded in the compressed data stream. Only the visible imagery 8 and 12-bit tables in MIL-STD-188-198A are allowed as defaults. Applications are free to develop tables more appropriate to their imagery than those described here. Any such tables must be embedded in the data stream.

4.5 Interchange format-decoders. All decoders shall interpret the full and abbreviated interchange format.

4.6 Further general requirements. Further requirements regarding the NITFS JPEG lossy compression algorithm apply to this standard and may be found in MIL-STD-188-198A. In the event of a conflict between the text of this standard and MIL-STD-188-198A, this standard shall take precedence.

5. DETAILED REQUIREMENTS

5.1 Image downsampling process. Downsampling is the process of reducing the size of an image relative to the original. In this document, downsampling is performed by simultaneously filtering the image and selecting a subset of the total samples available from the original. The output image will have fewer pixels and reduced dimensions, but will still be recognizable as the original image. Furthermore, the downsampling is carried out such that the aspect ratio is consistent between the original and downsampled images. Inputs into the downsampling module are the original image and a downsample ratio that relates the total number of pixels in the original image to the downsampled image. The output of this module is delivered to a NITFS compliant JPEG module for further compression and formation of a coded bitstream.

The calculation of the downsampled image dimensions and adjusted downsample ratios are discussed in Section 5.1.1. The mechanics of the one-dimensional filtering operation are explained in Section 5.1.2, while the necessary equations to calculate the filter parameters are given in Section 5.1.2.1. Section 5.1.3 describes in general how the filtering operation is applied to images.

5.1.1 Downsampled image dimensions. The downsampled image will have reduced dimensions with respect to the original image. The number of rows and columns of the downsampled image must be calculated separately in order to properly account for non-square images. The two dimensions are calculated using the following equations.

Downsampled image rows:

$$= \left(\frac{N_o}{R_o} \right)$$

Downsampled image columns:

$$= \left(\frac{M_o}{R_o} \right)$$

where N_o and M_o are the number of rows and columns in the original image respectively, and R_o is the original downsample ratio. The actual downsample ratio that is to be used in subsequent processing should be altered to reflect the fact that truncation is performed to obtain the downsampled image dimensions. For this situation, the downsample ratio must be calculated separately for the row and columns dimensions.

Downsample ratio for the row dimension:

$$= \frac{N_o}{N_d}$$

Downsample ratio for the column dimension:

= —

5.1.2 Downsampling filter operation. The downsampled image is formed by performing separable one-dimensional filtering on the rows and columns of the original image. The filtering operation is described in the following equation as the weighted average of samples.

Filtering equation for downsampling:

$$y_i = \sum_{j=e_i}^{b_i} w_{ij} x_j$$

where y_i denotes a sample in the output image, x_j denotes a sample in the input image, b_i and e_i specify integer limits on the summations, and w_{ij} is the filter coefficient associated with output sample, i , and input sample, j . When filtering is performed in the row dimension, then y_i and x_j refer to row samples; when filtering is performed in the column dimension, then y_i and x_j refer to column samples. The equation is applied similarly for all elements in a single dimension using the same set of parameters, e_i , b_i , and w_{ij} , so no designation has been made for the particular row or column that is filtered. However, the integer limits and the filter coefficients must be calculated separately for the rows and columns when the image is non-square. The filtering operation is illustrated in Figure 4 for the row processing case with $i = 10$, $b_{10} = 11$, $e_{10} = 15$, and a downsample ratio, $R = 1.3$.

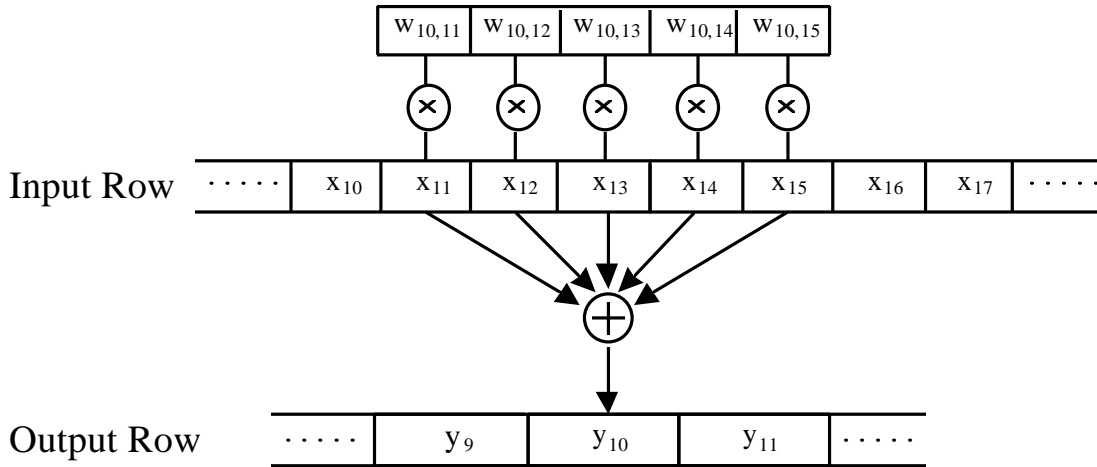


FIGURE 4. Downsampling demonstration with $i=10$, $b_{10}=11$, $e_{10}=15$, and $R = 1.3$.

5.1.2.1 Downsample filter parameter calculations. The pertinent parameters that are required for implementation of the downsampling filter are the integer summation limits, b_i and e_i , and the coefficients, w_{ij} . The calculation of these parameters differs slightly depending on the dimension that is considered, due to the change in downsample ratio as discussed in

Section 5.1.1. However, one set of parameters can be used for all the elements in the associated dimension (e.g. one set of row parameters can be applied to all the rows).

The summation limits can be calculated as shown in the following equations.

Filter beginning index:

$$= (b - a \cdot \quad)$$

Filter ending index:

$$= (b + a \cdot \quad)$$

where:

$$\begin{aligned} a &= \\ &= \left\{ \right. \\ b &= \cdot + \quad \cdot - \end{aligned}$$

The parameter, α , is a value that specifies a fixed filter length, while R refers to the downsample ratios discussed in Section 5.1.1. b_i is a variable describing the location of the filter center relative to the input samples.

The filter coefficients, w_{ij} , can be calculated in a two-step process.

Filter coefficients:

$$= \frac{\quad}{\sum_{\quad}}$$

where:

$$= \sqrt{\left(\frac{p \cdot (b - \quad)}{a \cdot \quad} \right) \times \left(\frac{p \cdot (b - \quad)}{\quad} \right)}$$

and:

$$= \left\{ \begin{array}{l} \frac{\quad}{\quad} \neq \\ \frac{\quad}{\quad} = \end{array} \right.$$

5.1.3 Application of the downsampling filter. One-dimensional filtering is applied repeatedly along each dimension until all samples in the downsampled image have been computed. Filtering along each dimension is performed independently. One dimension is processed entirely before continuing to the complementary dimension. After processing one dimension, an intermediate image is formed as the input for processing in the other dimension.

Note that the processing order (e.g. rows then columns, or vice versa) can be chosen so as to maximize performance for a given system platform. These concepts are further described in Figure 6, which shows the control procedure for the downsampling operation for the example of row-column order processing.

5.1.3.1 Downsampling along the image edges. In the course of downsampling an image, input values are needed that lie outside the original image. This occurs at the top, bottom, left, and right edges of the image. When extra data is needed, enough samples shall be generated by mirroring values from within the image so that the filter coefficients will always coincide with actual image samples. The mirroring point coincides with the input data sample that is exactly on the edge (e.g. first sample in a row when padding on the left of the image). Therefore, the edge sample is never repeated. This is illustrated in Figure 5.

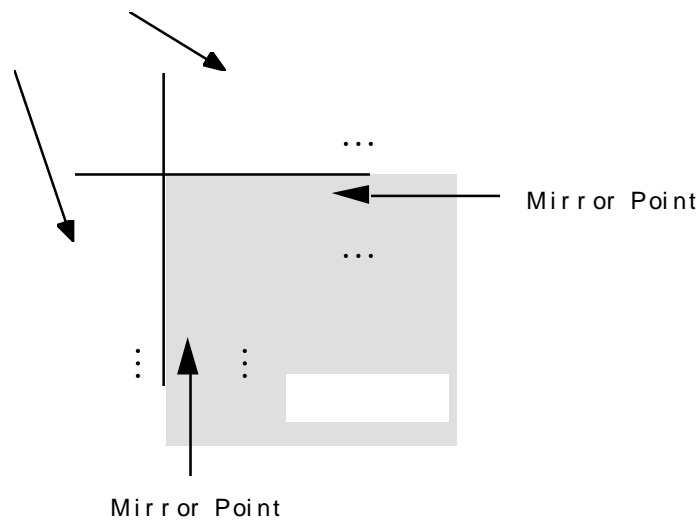
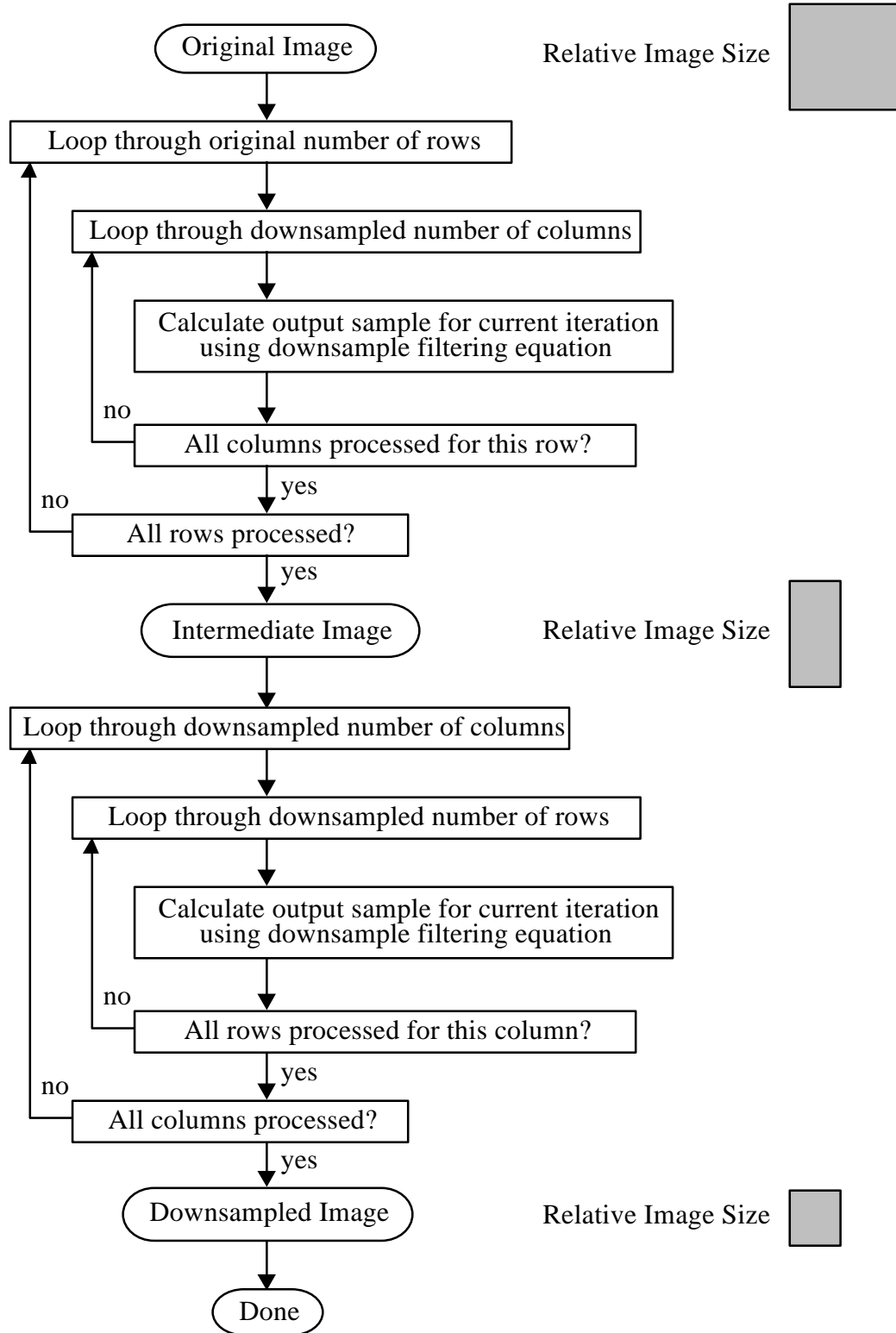


FIGURE 5. Illustration of mirroring for image edges.

FIGURE 6. Control procedure for image downsampling (row-column order).

5.2 JPEG compression of the downsampled image. The requirements and control procedures pertaining to the sequential DCT-based JPEG mode in MIL-STD-188-198A apply to this standard except as noted below. Once downsampling of the original image data is completed, the resultant downsampled image data is compressed with the NITFS JPEG sequential DCT mode image compression algorithm. Changes in the compressed image data format are described in the following sections. The appropriate flags and parameter values for relevant fields in the NITF image subheader are given for the interim algorithm in section 5.2.2.2. Suggested quantization and Huffman tables for both 8-bit and 12-bit gray scale imagery may be found in Appendix A.

5.2.1 Control procedures for the sequential DCT mode. The control procedures for encoding an image using the JPEG sequential DCT mode may be found in IS 10918-1. It is required by this standard that an NITF APP₆ “NITF” application data segment be placed in the compressed data stream. This data segment immediately follows the first SOI marker in the Image Data Field (see Figure 7). The format and content of this data segment are discussed in section 5.2.2.3. Additional requirements and control procedures for NITFS JPEG sequential DCT mode may be found in MIL-STD-188-198A .

5.2.2 Compressed data interchange format. The interchange format consists of an ordered collection of markers, parameters, and entropy-coded data segments. A detailed description of the format is given in MIL-STD-188-198A. In the following sections the changes to this interchange format necessary when using the interim algorithm are given.

5.2.2.1 Format of a JPEG compressed image within an NITF file. The format for NITF image data compressed with the sequential lossless JPEG mode differs based on the number of blocks, bands, and IMODE value (B, P, see MIL-STD-2500A). These different cases are described below. Note that IMODE = S is not appropriate for an interim algorithm multiple-block file since this standard is single band (8-bit and 12-bit gray scale) in nature.

5.2.2.1.1 Single block JPEG compressed format. The format for NITF single block image data compressed with the sequential lossless JPEG mode is shown in Figure 7.

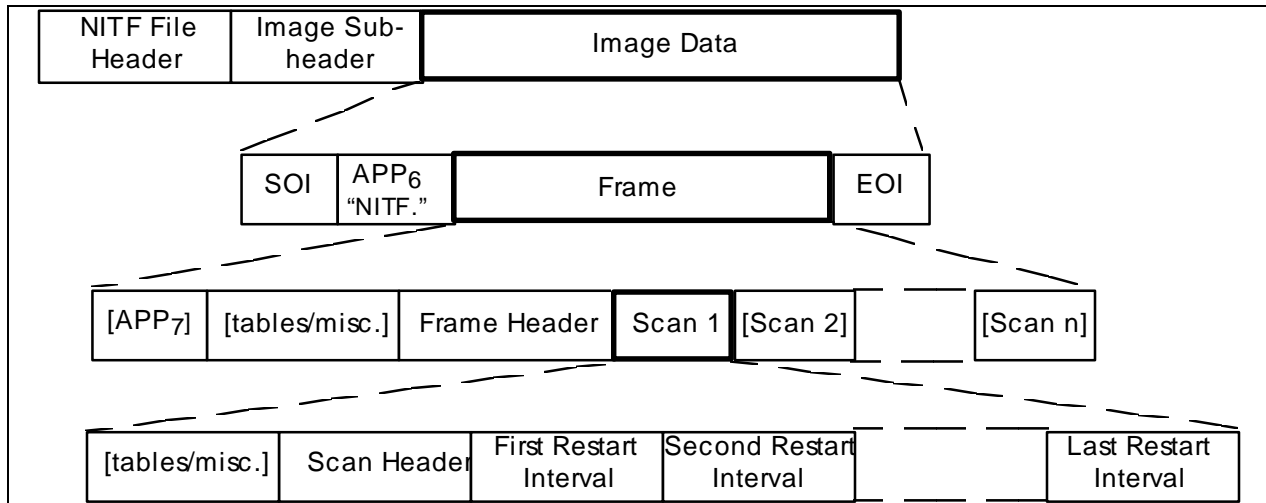


FIGURE 7. NITF single block file structure (IMODE=B or P).

5.2.2.1.1.1 Single block image data format. The top level of Figure 7 specifies that the JPEG compressed data is contained in the Image Data Field of the NITF file. The second level of Figure 7 specifies that the single block image format shall begin with an SOI marker, shall contain one frame, and shall end with an EOI marker. Between the SOI/EOI marker pair, the data stream is compliant with IS 10918-1 subject to the requirements and constraints of this standard.

5.2.2.1.1.2 Frame and Scan formats. The frame and scan marker formats in Figure 7 are the same as those found in MIL-STD-188-198A. The Start-of-Frame (SOF) marker data segment contains two fields “Y” and “X” which contain the number of lines and the number of samples per line in the compressed image. For the interim low bit-rate algorithm, these fields shall contain the number of lines and the number of samples per line for the downsampled image data. These fields must reflect the size of the image that underwent JPEG compression.

5.2.2.1.2 Multiple block JPEG compressed format. The format for NITF multiple block image data compressed with the sequential lossless JPEG mode is shown in Figure 8.

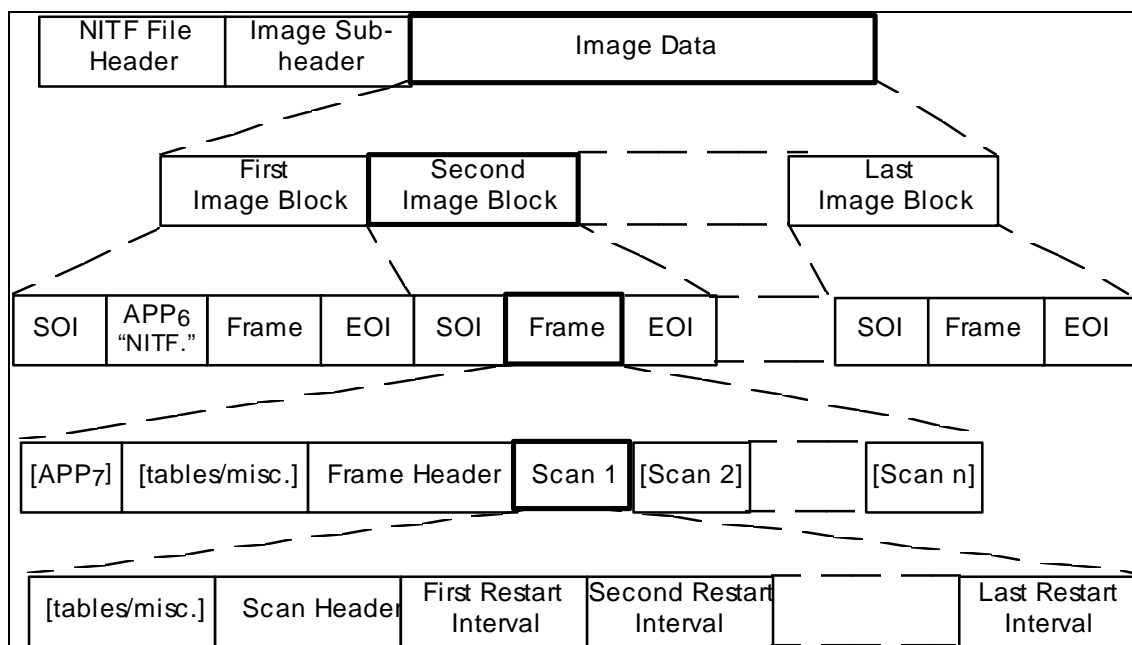


FIGURE 8. NITF multiple block file structure (IMODE=B or P).

5.2.2.1.2.1 Multiple block image data format. When the interim algorithm is applied to a blocked image, pre and postprocessing are performed on a block by block basis prior to JPEG compression of each block. All blocks therefore have identical horizontal and vertical downsampling ratios due to the NITF restriction that all blocks have the same horizontal dimension and the same vertical dimension.

The top level of Figure 8 specifies that the JPEG compressed data is contained in the Image Data Field of the NITF file. The second level of Figure 8 specifies that this multiple block image format shall begin with the compressed data for the first image block and shall be followed by the compressed data for each image block, one after the other, left to right, top to bottom. The third level of Figure 8 specifies that each compressed block shall begin with an SOI marker, shall contain one frame, and shall end with an EOI marker. A required application marker data segment immediately follows the SOI marker in the first image block. This APP₆ marker data segment is described in section 5.2.2.3.

5.2.2.1.2.2 Frame and Scan formats. The frame and scan marker formats in Figure 5 are the same as those found in MIL-STD-188-198A. The SOF marker data segment must reflect the downsampled image data size.

5.2.2.2 NITF image subheader. Fields in the NITF image subheader must reflect the original image size. The interim algorithm is unique in that the image and block sizes in the NITF image subheader do not match the image or block sizes in the JPEG SOF marker data segment(s). This is necessary since the JPEG compression operates on the downsampled image or blocks while ancillary NITF data such as overlays apply only to the original image or block dimensions.

The NROWS and NCOLS fields of the image subheader shall contain the number of significant rows and columns, respectively, in the original image. The NPPBH and NPPBV fields shall contain the number of pixels per block horizontal and the number of pixels per block vertical, respectively, of the original blocks in a blocked image.

The IMAG field of the NITF image subheader is not modified for the interim algorithm. Any decoder capable of decoding an interim algorithm compressed data file must restore the image and blocks to their original dimensions.

The COMRAT field of the NITF image subheader is set to 00.0 when using the recommended tables in Appendix A or any other non-default tables. This setting indicates general purpose imagery and embedded tables. If an application uses default tables, this is indicated in the COMRAT field by $XX.Y \in [00.1, 00.2, 00.3, 00.4, 00.5]$ where $XX = 00$ indicates general purpose imagery and $Y \in [1, 2, 3, 4, 5]$ indicates the quality level. The default quantization and Huffman tables for 8-bit and 12-bit imagery may be found in MIL-STD-188-198A. The specific tables chosen are those for visible imagery. It is intended that future NITF compliant systems shall always embed their tables in the compressed data stream. Implementors are strongly encouraged to embed quantization and Huffman tables.

5.2.2.3 APP₆ “NITF” application data segment. NITF requires the use of an APP₆ “NITF” application data segment. This application data segment shall immediately follow the first SOI marker in the image data field. The “NITF” application data segment contains information which is needed by an interpreter but not supported by the ISO/CCITT JPEG format.

TABLE I. APP₆ “NITF” application data segment.

Offset	Field Value	Field Name	length (bytes)	comments
0	0xFFE6	APP ₆	2	NITF application data marker.
2	25	L _p	2	Segment length (2+length of application data)
4	0x4E49 0x5446 0x00	Identifier	5	Null terminated string: "NITF"
9	0x0200	Version	2	Version number. The most significant byte is used for major revisions, the least significant byte for minor revisions. Version 2.00 is the current revision level.
11	0x42, 0x50 or 0x53	IMODE	1	Image Format. Two values are defined at this time. ‘B’ - IMODE=B ‘P’ - IMODE=P
12	1-9999	H	2	Number of image blocks per row.
14	1-9999	V	2	Number of image blocks per column.
16	0-1	Image Color	1	Original image color representation. One value is defined at this time. 0 - monochrome
17	1-16	Image Bits	1	Original image sample precision.
18	0-99	Image Class	1	Image data class (0-99). One value is defined at this time 0 - general purpose
19	1 - 29	JPEG Process	1	JPEG coding process. The values for this field are defined to be consistent with ISO IS 10918-2. 1 - baseline sequential DCT, Huffman coding, 8-bit sample precision 4 - extended sequential DCT, Huffman coding, 12-bit sample precision
20	0-5	Quality	1	Image default quantization tables used. Quality values 1-5 select specific default tables. The value 0 indicates no defaults and all quantization tables must then be present in the stream.
21	0-2	Stream Colour	1	Compressed colour representation. One value is defined at this time. 0 - monochrome
22	8 or 12	Stream Bits	1	Compressed image sample precision.
23	0	Flags	4	Reserved for future use.

5.3 JPEG decompression of the downsampled image. Prior to upsampling, JPEG decompression takes place resulting in a lossy reconstruction of the downsampled image data. The control procedures for decoding an image compressed with the JPEG sequential DCT

mode may be found in IS 10918-1. These procedures are to be followed pursuant to the requirements of MIL-STD-188-198A.

5.4 Image upsampling process. Upsampling is the process of increasing the number of samples through interpolation of the existing values. The process is very similar to downsampling as described in Section 5.1, but in this case we will be sampling the image more frequently to increase the number of image samples. Filtering is applied to the downsampled, reconstructed image that is received from the NITFS-compliant JPEG reconstruction module. The filtering operation generates enough new samples so that the upsampled image dimensions will match the dimensions of the original image.

Calculation of the upsample ratios is discussed in Section 5.4.1. The mechanics of the one-dimensional filtering operation are explained in Section 5.4.2, while the necessary equations to calculate the filter parameters are given in Section 5.4.2.1. Section 5.4.3 describes in general how the filtering operation is to be applied to images.

5.4.1 Upsample ratio calculation. Separate upsample ratios must be calculated for each dimension. The two upsample ratios define the amount of expansion that is required in order to match the resolution of the downsampled image to the original image. The ratios are only dependent on the number of rows and columns in the original image, specified by N_O and M_O , and the downsampled image, specified by N_d and M_d .

Upsample ratio for the row dimension:

$$= \frac{N_O}{N_d}$$

Upsample ratio for the column dimension:

$$= \frac{M_O}{M_d}$$

Note that the ratios are equivalent to the downsample ratios shown in Section 5.1.1.

5.4.2 Upsampling filter operation. The upsampled image is formed by performing separable one-dimensional filtering on the rows and columns of the downsampled image. The mechanics of the upsample filtering process is exactly the same as the downsample case with the exception of parameter calculation. The following equation is equivalent to the filtering equation for downsampling found in Section 5.1.2, but repeated here for convenience.

Filtering equation for upsampling:

$$= \sum_{n=-N}^{N} \cdot$$

where y_i denotes a sample in the output image, x_j denotes a sample in the input image, b_i and e_i specify integer limits on the summations, and w_{ij} is the filter coefficient associated with output sample, i , and input sample, j . When filtering is performed in the row dimension, then y_i and x_j refer to row samples; when filtering is performed in the column dimension, then y_i and x_j refer to column samples. The equation is applied similarly for all elements in a single dimension using the same set of parameters, e_i , b_i , and w_{ij} , so no designation has been made for the particular row or column that is filtered. However, the integer limits and the filter coefficients must be calculated separately for the rows and columns when the image is non-square. The filtering operation illustrated in figure 4 for the row processing case with $i = 10$, $b_{10} = 6$, $e_{10} = 9$, and an upsample ratio, $R = 1.3$.

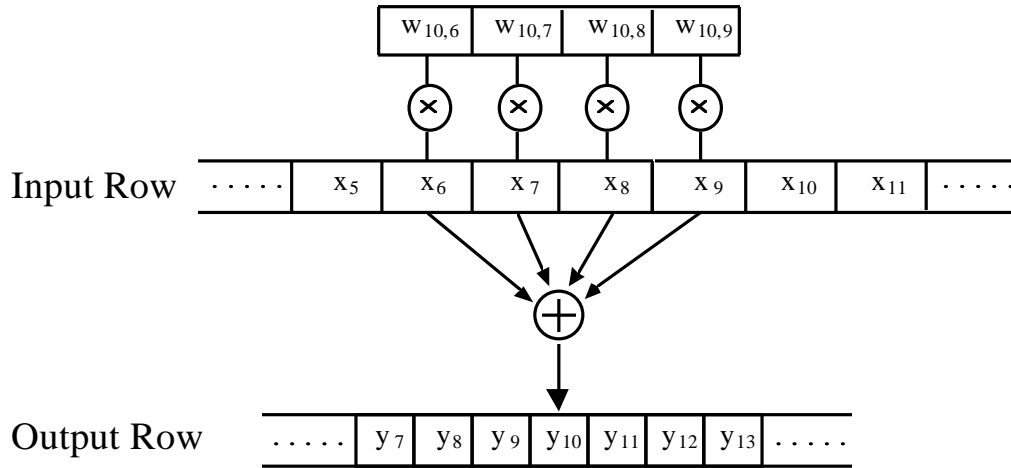


FIGURE 9. Upsampling demonstration with $i=10$, $b_{10}=6$, $e_{10}=9$, and $R = 1.3$.

5.4.2.1 Upsample filter parameter calculations. Similar to downsampling, the parameters that require calculation are the integer summation limits and filter coefficients. One set of parameters can be applied for all the elements in the associated dimension (e.g. one set of row parameters can be applied to all the rows).

Filter beginning index:

$$= (b - a)$$

Filter ending index:

$$= (b + a)$$

where:

$$\begin{aligned} a &= \\ &= \left\{ \begin{array}{l} \cdot \\ \cdot \end{array} \right. \\ b &= \cdot - \cdot + \end{aligned}$$

The parameter, α , is a value that specifies a fixed filter length, while R refers to the upsample ratios discussed in Section 5.4.1. b_i is a variable describing the location of the filter center relative to the input samples. (Refer to Section 6.0 for more information concerning the upsample filter length).

The filter coefficients, w_{ij} , can be calculated in a two-step process.

Filter coefficients:

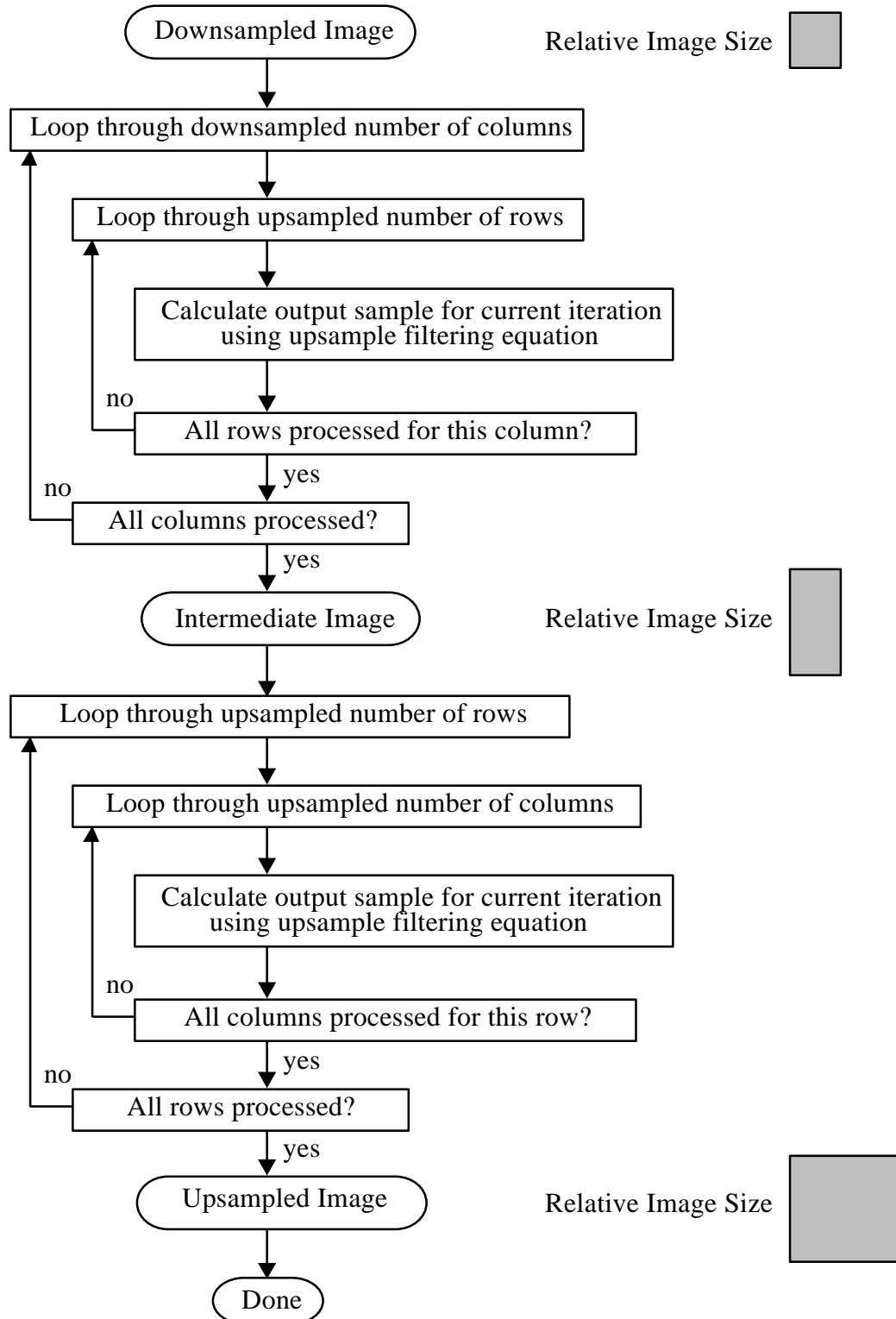
$$= \frac{\sum_{i=0}^{\alpha-1} w_{ij}}{\alpha}$$

where:

$$= \left(\left(\frac{p \cdot (b - i)}{a} \right) \right) \times (p \cdot (b - i))$$

5.4.3 Application of the upsampling filter. One-dimensional filtering is applied repeatedly along each dimension until all samples in the upsampled image have been computed. Filtering along each dimension is performed independently. One dimension is processed entirely before continuing to the complementary dimension. After processing one dimension, an intermediate image is formed as the input for processing in the other dimension. Note that the processing order (e.g. rows then columns, or vice versa) can be chosen so as to maximize performance for a given system platform. These concepts are further described in Figure 10, which shows the general procedure for the upsampling operation for the example of column-row order processing.

5.4.3.1 Upsampling along the image edges. In the course of upsampling an image, input values are needed that lie outside the sampling grid of the downsampled image. This occurs at the top, bottom, left, and right edges of the image. When extra data is needed, enough samples shall be generated by mirroring values from within the image so that the filter coefficients will always coincide with actual image samples. The mirroring point coincides with the input data sample that is exactly on the edge (e.g. first sample in a row when padding on the left of the image). Therefore, the edge sample is never repeated. This is illustrated in Figure 5 found in Section 5.1.3.1.

FIGURE 10. Control procedure for image upsampling (row-column order).

6. NOTES

This section is informative only. Comments, explanations, and warnings about the compression system defined in this document are given as ancillary information. The formal requirements are outlined in Sections 4.0 and 5.0.

6.1 Downsample ratio calculation. Since downsampled image data will be compressed with JPEG, greater coding efficiency can be obtained by tuning the downsampling ratio to create downsampled image dimensions which are integer multiples of 8. This prevents the JPEG algorithm from padding blocks at image edges and wasting bit-rate. The following equations calculate the proper downsampled image dimensions for maximal JPEG coding efficiency. This set of equations can be substituted for those defined in Section 5.1.1.

Downsampled image rows incorporating JPEG block size:

$$= \left\lceil \frac{N_o}{R_o} \right\rceil$$

Downsampled image columns incorporating JPEG block size:

$$= \left\lceil \frac{M_o}{R_o} \right\rceil$$

where N_o and M_o are the number of rows and columns in the original image respectively, and R_o is the original downsample ratio.

6.2 Image upsampling. Certain applications may require faster upsampling at the cost of much reduced image quality. This is typically the case when computational resources are limited. Bilinear interpolation is often identified as being suitable for these applications since each interpolated value is a function of only the four nearest pixels in two-dimensions, which is a significant decrease in complexity. Furthermore, standard software routines and dedicated hardware exists to perform fast bilinear interpolation. The danger of using a standard package to perform upsampling is that the result will be a shifted version of what is expected. This is due to a phase shift in the downsampled image described by the following equation.

Offset in the downsampled image:

$$= \frac{R_o}{R} - 1$$

where R is the upsample ratio discussed in Section 5.4.2.1. To ensure proper decoding, the offset must be removed during the upsampling process. The recommended method to reduce the complexity of the decoder while maintaining the proper pixel sampling positions is to

reduce the filter length parameter from the value of four to two. It must be emphasized that the quality of the decoded image is noticeably worse than nominal when bilinear interpolation is used. The tradeoff between complexity and quality must be evaluated carefully before deciding to reduce the length of the upsampling filter.

6.3 Overlays. Overlays for the decoded images are intolerant to changes in image size. To prevent difficulties with overlays, the upsampled image is constrained to have dimensions equivalent to the original, uncompressed image, and the upsampling algorithm should conform to the specifications outlined in Section 5.4. If the decoder does not properly upsample the image, the overlays will be placed at incorrect locations in the image. Such an error could have serious implications for imagery users.

6.4 Inherent quality losses. The compression system specified in this document is aimed towards applications requiring very low bit rate compression. The nature of this requirement dictates that much information will be lost in the compression process, albeit minimization of this loss is the objective of the algorithm design. Therefore, visible distortions and resolution degradation are to be expected in the resulting images (e.g. NIIRS losses greater than 1.0 are not uncommon). It should be emphasized that this compressor is not meant for high quality compression of images.

6.5 Incompatibility issues with previous generations of NITFS systems. Historical NITFS systems will not have the capability to decode images that have been compressed using the algorithm specified in this document. A new image compression type (IC field in NITF image subheader) has been created to handle the lack of backwards compatibility. Previous generation systems will be unable to decode these images since they will not recognize this new compression type. The IC field value, I1, was for this purpose. Previous systems could conceivably decode the JPEG compressed downsampled image data (see Figure 2), but be unable to upsample the data to its proper size. As noted in Section 6.3, this could have serious consequences if overlay information is present in the image product.

APPENDIX A

RECOMMENDED COMPRESSION PARAMETERS

A.1 Recommended compression parameters. The recommended compression parameters allows images to be coded at five different quality levels, which will be referred to as IQ1, IQ2, IQ3, IQ4, and IQ5. IQ5 (quality level 5) compression has the highest fidelity to the source image, but achieves the least compression. IQ1 compression results in the worst reconstructed image quality, but the highest compression. The IQ2, IQ3, and IQ4 levels represent compromises between IQ1 and IQ5 in ascending order of quality, and descending order of compression.

In the following sections, the compression parameters are given for both 8-bit and 12-bit source images. The pertinent parameters for each quality level are the downsample ratio (Refer to Section 5.1), JPEG quantization table (Refer to MIL-STD-188-198A), and the JPEG Huffman tables (Refer to MIL-STD-188-198A). Please note that these parameters are currently not fully optimized. Additionally, the parameters were developed only for the eight-bit and twelve-bit classes of visible imagery. Parameters which are fully optimized across a wide range of image classes may become available in a future version of this document.

A.2 Eight-bit gray scale compression parameters.

A.2.1 Downsample ratios. The downsample ratios for each quality level is shown in the following table. The downsample ratio at a particular quality level is applied as specified in Section 5.1.

TABLE A1. Downsample ratios for eight-bit gray scale images .

Quality Level	Downsample Ratio
IQ1	27.0
IQ2	13.5
IQ3	7.5
IQ4	4.5
IQ5	1.5

A.2.2 JPEG quantization table. The following quantization table applies to all quality levels (IQ1-IQ5). The values are formatted as an 8x8 matrix of quantization scale factors. The format is such that the matrix entry at a particular index location is the quantization factor to be applied to the DCT frequency coefficient at the matching index location. For example, the first element, located in the top-left position, is used to quantize the DC DCT frequency coefficient.

TABLE A2. Eight-bit gray scale JPEG quantization table .

36	36	37	39	42	45	50	54
36	37	39	42	45	50	54	60
37	39	42	45	50	54	60	66
39	42	45	50	54	60	66	74
42	45	50	54	60	66	74	81
45	50	54	60	66	74	81	90
50	54	60	66	74	81	90	99
54	60	66	74	81	90	99	110

A.2.3 JPEG Huffman tables. The parameters defined in Sections A.2.3.1 through A.2.3.5 specify the DC and AC BITS and HUFFVAL tables needed to generate Huffman codes for the pre-defined quality levels. BITS and HUFFVAL are described in the JPEG standard document, MIL-STD-188-198A. The "luminance" label is to emphasize that these tables are meant for the coding of gray scale images.

A.2.3.1 IQ1 Huffman table parameters.

dc_luminance_bits[16] = 0, 3, 1, 1, 1, 1, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0

dc_luminance_val[12] = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

ac_luminance_bits[16] = 0, 2, 2, 1, 3, 3, 2, 4, 6, 1, 1, 1, 1, 0, 0, 135

ac_luminance_val[162] = 0, 1, 2, 17, 33, 3, 49, 65, 18, 81,
97, 113, 129, 19, 34, 145, 161, 4, 50, 177,
193, 209, 240, 66, 225, 82, 241, 20, 35, 51,
98, 5, 114, 130, 146, 67, 83, 162, 21, 52,
210, 36, 37, 178, 6, 7, 8, 9, 10, 22,
23, 24, 25, 26, 38, 39, 40, 41, 42, 53,
54, 55, 56, 57, 58, 68, 69, 70, 71, 72,
73, 74, 84, 85, 86, 87, 88, 89, 90, 99,
100, 101, 102, 103, 104, 105, 106, 115, 116, 117,
118, 119, 120, 121, 122, 131, 132, 133, 134, 135,
136, 137, 138, 147, 148, 149, 150, 151, 152, 153,
154, 163, 164, 165, 166, 167, 168, 169, 170, 179,
180, 181, 182, 183, 184, 185, 186, 194, 195, 196,
197, 198, 199, 200, 201, 202, 211, 212, 213, 214,
215, 216, 217, 218, 226, 227, 228, 229, 230, 231,
232, 233, 234, 242, 243, 244, 245, 246, 247, 248,
249, 250

A.2.3.2 IQ2 Huffman table parameters.

```

dc_luminance_bits[16] = 0, 3, 1, 1, 1, 1, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0
dc_luminance_val[12] = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
ac_luminance_bits[16] = 0, 1, 3, 3, 3, 2, 3, 8, 2, 1, 1, 1, 1, 0, 2, 131
ac_luminance_val[162] = 1, 0, 2, 17, 3, 33, 49, 18, 65, 81,
                        97, 113, 34, 129, 145, 4, 19, 50, 161, 177,
                        193, 209, 240, 66, 225, 82, 241, 20, 35, 98,
                        5, 51, 114, 146, 67, 130, 83, 162, 21, 36,
                        52, 178, 99, 115, 147, 194, 210, 6, 7, 8,
                        9, 10, 22, 23, 24, 25, 26, 37, 38, 39,
                        40, 41, 42, 53, 54, 55, 56, 57, 58, 68,
                        69, 70, 71, 72, 73, 74, 84, 85, 86, 87,
                        88, 89, 90, 100, 101, 102, 103, 104, 105, 106,
                        116, 117, 118, 119, 120, 121, 122, 131, 132, 133,
                        134, 135, 136, 137, 138, 148, 149, 150, 151, 152,
                        153, 154, 163, 164, 165, 166, 167, 168, 169, 170,
                        179, 180, 181, 182, 183, 184, 185, 186, 195, 196,
                        197, 198, 199, 200, 201, 202, 211, 212, 213, 214,
                        215, 216, 217, 218, 226, 227, 228, 229, 230, 231,
                        232, 233, 234, 242, 243, 244, 245, 246, 247, 248,
                        249, 250

```

A.2.3.3 IQ3 Huffman table parameters.

```

dc_luminance_bits[16] = 0, 3, 1, 1, 1, 1, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0
dc_luminance_val[12] = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
ac_luminance_bits[16] = 0, 1, 3, 3, 3, 2, 3, 8, 2, 1, 1, 1, 1, 0, 2, 131
ac_luminance_val[162] = 1, 0, 2, 17, 3, 33, 49, 18, 65, 81,
                        97, 113, 34, 129, 145, 4, 19, 50, 161, 177,
                        193, 209, 240, 66, 225, 82, 241, 35, 20, 51,
                        98, 114, 130, 146, 5, 83, 67, 162, 21, 99,
                        36, 52, 178, 210, 6, 7, 8, 9, 10, 22,
                        23, 24, 25, 26, 37, 38, 39, 40, 41, 42,
                        53, 54, 55, 56, 57, 58, 68, 69, 70, 71,
                        72, 73, 74, 84, 85, 86, 87, 88, 89, 90,
                        100, 101, 102, 103, 104, 105, 106, 115, 116, 117,
                        118, 119, 120, 121, 122, 131, 132, 133, 134, 135,
                        136, 137, 138, 147, 148, 149, 150, 151, 152, 153,
                        154, 163, 164, 165, 166, 167, 168, 169, 170, 179,
                        180, 181, 182, 183, 184, 185, 186, 194, 195, 196,
                        197, 198, 199, 200, 201, 202, 211, 212, 213, 214,
                        215, 216, 217, 218, 226, 227, 228, 229, 230, 231,
                        232, 233, 234, 242, 243, 244, 245, 246, 247, 248,
                        249, 250

```

A.2.3.4 IQ4 Huffman table parameters.

```

dc_luminance_bits[16] = 0, 3, 1, 1, 1, 1, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0
dc_luminance_val[12] = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
ac_luminance_bits[16] = 0, 1, 3, 3, 3, 2, 3, 8, 2, 1, 1, 1, 1, 0, 2, 131
ac_luminance_val[162] = 1, 0, 2, 17, 3, 33, 49, 18, 65, 81,
                        97, 113, 34, 129, 145, 4, 19, 50, 161, 177,
                        193, 209, 240, 66, 225, 82, 241, 35, 20, 51,
                        98, 114, 146, 5, 67, 130, 162, 83, 178, 36,
                        99, 21, 52, 115, 210, 6, 7, 8, 9, 10,
                        22, 23, 24, 25, 26, 37, 38, 39, 40, 41,
                        42, 53, 54, 55, 56, 57, 58, 68, 69, 70,
                        71, 72, 73, 74, 84, 85, 86, 87, 88, 89,
                        90, 100, 101, 102, 103, 104, 105, 106, 116, 117,
                        118, 119, 120, 121, 122, 131, 132, 133, 134, 135,
                        136, 137, 138, 147, 148, 149, 150, 151, 152, 153,
                        154, 163, 164, 165, 166, 167, 168, 169, 170, 179,
                        180, 181, 182, 183, 184, 185, 186, 194, 195, 196,
                        197, 198, 199, 200, 201, 202, 211, 212, 213, 214,
                        215, 216, 217, 218, 226, 227, 228, 229, 230, 231,
                        232, 233, 234, 242, 243, 244, 245, 246, 247, 248,
                        249, 250

```

A.2.3.5 IQ5 Huffman table parameters.

```

dc_luminance_bits[16] = 0, 3, 1, 1, 1, 1, 0, 2, 3, 0, 0, 0, 0, 0, 0, 0
dc_luminance_val[12] = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
ac_luminance_bits[16] = 0, 1, 3, 2, 5, 2, 3, 7, 4, 1, 1, 1, 1, 1, 1, 129
ac_luminance_val[162] = 1, 0, 2, 17, 33, 49, 3, 18, 65, 81,
                        97, 113, 129, 34, 145, 161, 4, 19, 50, 177,
                        193, 209, 240, 66, 82, 225, 241, 35, 20, 51,
                        98, 114, 130, 146, 5, 162, 21, 52, 67, 83,
                        178, 6, 7, 8, 9, 10, 22, 23, 24, 25,
                        26, 36, 37, 38, 39, 40, 41, 42, 53, 54,
                        55, 56, 57, 58, 68, 69, 70, 71, 72, 73,
                        74, 84, 85, 86, 87, 88, 89, 90, 99, 100,
                        101, 102, 103, 104, 105, 106, 115, 116, 117, 118,
                        119, 120, 121, 122, 131, 132, 133, 134, 135, 136,
                        137, 138, 147, 148, 149, 150, 151, 152, 153, 154,
                        163, 164, 165, 166, 167, 168, 169, 170, 179, 180,
                        181, 182, 183, 184, 185, 186, 194, 195, 196, 197,
                        198, 199, 200, 201, 202, 210, 211, 212, 213, 214,
                        215, 216, 217, 218, 226, 227, 228, 229, 230, 231,
                        232, 233, 234, 242, 243, 244, 245, 246, 247, 248,
                        249, 250

```

A.3 Twelve-bit gray scale compression parameters.

A.3.1 Downsample ratios. The downsample ratios for each quality level is shown in the following table. The downsample ratio at a particular quality level is applied as specified in Section 5.1. Note that the downsample ratios for twelve-bit images are the same as for eight-bit images.

TABLE A3. Downsample ratios for twelve-bit gray scale images .

Quality Level	Downsample Ratio
IQ1	27.0
IQ2	13.5
IQ3	7.5
IQ4	4.5
IQ5	1.5

A.3.2 JPEG quantization table. The following quantization table applies to all quality levels (IQ1-IQ5). The format of the table shown below follows the format used to describe the eight-bit gray scale JPEG quantization table in Section A.2.2.

TABLE A4. Twelve-bit gray scale JPEG quantization table .

576	576	592	624	672	720	800	864
576	592	624	672	720	800	864	960
592	624	672	720	800	864	960	1056
624	672	720	800	864	960	1056	1184
672	720	800	864	960	1056	1184	1296
720	800	864	960	1056	1184	1296	1440
800	864	960	1056	1184	1296	1440	1584
864	960	1056	1184	1296	1440	1584	1760

A.3.3 JPEG Huffman tables. The parameters defined in Sections A.3.3.1 through A.3.3.5 specify the DC and AC BITS and HUFFVAL tables needed to generate Huffman codes for the pre-defined quality levels. These tables apply to compression of twelve-bit imagery only. An explanation of the format used to specify the Huffman parameters can be found in Section A.2.3.

A.3.3.1 IQ1 Huffman table generation parameters.

dc_luminance_bits[16] = 0, 3, 1, 1, 1, 1, 0, 0, 6, 3, 0, 0, 0, 0, 0, 0

dc_luminance_val[16] = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

ac_luminance_bits[16] = 0, 2, 2, 1, 3, 3, 2, 4, 6, 0, 1, 1, 0, 2, 0, 199

ac_luminance_val[226] = 0, 1, 2, 17, 33, 3, 49, 65, 18, 81,
 97, 113, 129, 4, 19, 34, 145, 50, 161, 177,
 193, 209, 240, 66, 225, 82, 241, 20, 35, 51,
 98, 114, 130, 146, 5, 67, 83, 162, 21, 36,
 178, 210, 99, 6, 7, 8, 9, 10, 11, 12,
 13, 14, 22, 23, 24, 25, 26, 27, 28, 29,
 30, 37, 38, 39, 40, 41, 42, 43, 44, 45,
 46, 52, 53, 54, 55, 56, 57, 58, 59, 60,
 61, 62, 68, 69, 70, 71, 72, 73, 74, 75,
 76, 77, 78, 84, 85, 86, 87, 88, 89, 90,
 91, 92, 93, 94, 100, 101, 102, 103, 104, 105,
 106, 107, 108, 109, 110, 115, 116, 117, 118, 119,
 120, 121, 122, 123, 124, 125, 126, 131, 132, 133,
 134, 135, 136, 137, 138, 139, 140, 141, 142, 147,
 148, 149, 150, 151, 152, 153, 154, 155, 156, 157,
 158, 163, 164, 165, 166, 167, 168, 169, 170, 171,
 172, 173, 174, 179, 180, 181, 182, 183, 184, 185,
 186, 187, 188, 189, 190, 194, 195, 196, 197, 198,
 199, 200, 201, 202, 203, 204, 205, 206, 211, 212,
 213, 214, 215, 216, 217, 218, 219, 220, 221, 222,
 226, 227, 228, 229, 230, 231, 232, 233, 234, 235,
 236, 237, 238, 242, 243, 244, 245, 246, 247, 248,
 249, 250, 251, 252, 253, 254

A.3.3.2 IQ2 Huffman table generation parameters.

dc_luminance_bits[16] = 0, 3, 1, 1, 1, 1, 0, 0, 6, 3, 0, 0, 0, 0, 0, 0

dc_luminance_val[16] = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

ac_luminance_bits[16] = 0, 1, 3, 3, 3, 2, 3, 8, 2, 0, 1, 1, 1, 0, 1, 197

ac_luminance_val[226] = 1, 0, 2, 17, 3, 33, 49, 18, 65, 81,
 97, 113, 34, 129, 145, 4, 19, 50, 161, 177,
 193, 209, 240, 66, 225, 82, 241, 35, 98, 20,
 114, 51, 130, 146, 67, 5, 83, 162, 178, 210,
 21, 194, 6, 7, 8, 9, 10, 11, 12, 13,
 14, 22, 23, 24, 25, 26, 27, 28, 29, 30,
 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,
 46, 52, 53, 54, 55, 56, 57, 58, 59, 60,
 61, 62, 68, 69, 70, 71, 72, 73, 74, 75,
 76, 77, 78, 84, 85, 86, 87, 88, 89, 90,
 91, 92, 93, 94, 99, 100, 101, 102, 103, 104,
 105, 106, 107, 108, 109, 110, 115, 116, 117, 118,
 119, 120, 121, 122, 123, 124, 125, 126, 131, 132,
 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,

147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
 157, 158, 163, 164, 165, 166, 167, 168, 169, 170,
 171, 172, 173, 174, 179, 180, 181, 182, 183, 184,
 185, 186, 187, 188, 189, 190, 195, 196, 197, 198,
 199, 200, 201, 202, 203, 204, 205, 206, 211, 212,
 213, 214, 215, 216, 217, 218, 219, 220, 221, 222,
 226, 227, 228, 229, 230, 231, 232, 233, 234, 235,
 236, 237, 238, 242, 243, 244, 245, 246, 247, 248,
 249, 250, 251, 252, 253, 254

A.3.3.3 IQ3 Huffman table generation parameters.

dc_luminance_bits[16] = 0, 3, 1, 1, 1, 0, 0, 5, 5, 0, 0, 0, 0, 0, 0, 0

dc_luminance_val[16] = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

ac_luminance_bits[16] = 0, 1, 3, 3, 3, 2, 3, 8, 2, 0, 1, 1, 1, 0, 1, 197

ac_luminance_val[226] = 1, 0, 2, 17, 3, 33, 49, 18, 65, 81,
 97, 113, 34, 129, 145, 4, 19, 50, 161, 177,
 193, 209, 240, 66, 225, 241, 35, 82, 98, 20,
 114, 5, 21, 51, 130, 162, 226, 99, 146, 67,
 83, 36, 178, 194, 210, 6, 7, 8, 9, 10,
 11, 12, 13, 14, 22, 23, 24, 25, 26, 27,
 28, 29, 30, 37, 38, 39, 40, 41, 42, 43,
 44, 45, 46, 52, 53, 54, 55, 56, 57, 58,
 59, 60, 61, 62, 68, 69, 70, 71, 72, 73,
 74, 75, 76, 77, 78, 84, 85, 86, 87, 88,
 89, 90, 91, 92, 93, 94, 100, 101, 102, 103,
 104, 105, 106, 107, 108, 109, 110, 115, 116, 117,
 118, 119, 120, 121, 122, 123, 124, 125, 126, 131,
 132, 133, 134, 135, 136, 137, 138, 139, 140, 141,
 142, 147, 148, 149, 150, 151, 152, 153, 154, 155,
 156, 157, 158, 163, 164, 165, 166, 167, 168, 169,
 170, 171, 172, 173, 174, 179, 180, 181, 182, 183,
 184, 185, 186, 187, 188, 189, 190, 195, 196, 197,
 198, 199, 200, 201, 202, 203, 204, 205, 206, 211,
 212, 213, 214, 215, 216, 217, 218, 219, 220, 221,
 222, 227, 228, 229, 230, 231, 232, 233, 234, 235,
 236, 237, 238, 242, 243, 244, 245, 246, 247, 248,
 249, 250, 251, 252, 253, 254

A.3.3.4 IQ4 Huffman table generation parameters.

dc_luminance_bits[16] = 0, 3, 1, 1, 1, 0, 0, 5, 5, 0, 0, 0, 0, 0, 0, 0

dc_luminance_val[16] = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

ac_luminance_bits[16] = 0, 1, 3, 3, 3, 2, 3, 8, 1, 2, 1, 1, 1, 0, 2, 195

ac_luminance_val[226] = 1, 0, 2, 17, 3, 33, 49, 18, 65, 81,
 97, 113, 34, 129, 145, 4, 19, 50, 161, 177,
 193, 209, 240, 225, 66, 82, 241, 35, 98, 20,

51, 114, 130, 162, 226, 5, 21, 99, 146, 67,
 178, 36, 83, 115, 194, 210, 6, 7, 8, 9,
 10, 11, 12, 13, 14, 22, 23, 24, 25, 26,
 27, 28, 29, 30, 37, 38, 39, 40, 41, 42,
 43, 44, 45, 46, 52, 53, 54, 55, 56, 57,
 58, 59, 60, 61, 62, 68, 69, 70, 71, 72,
 73, 74, 75, 76, 77, 78, 84, 85, 86, 87,
 88, 89, 90, 91, 92, 93, 94, 100, 101, 102,
 103, 104, 105, 106, 107, 108, 109, 110, 116, 117,
 118, 119, 120, 121, 122, 123, 124, 125, 126, 131,
 132, 133, 134, 135, 136, 137, 138, 139, 140, 141,
 142, 147, 148, 149, 150, 151, 152, 153, 154, 155,
 156, 157, 158, 163, 164, 165, 166, 167, 168, 169,
 170, 171, 172, 173, 174, 179, 180, 181, 182, 183,
 184, 185, 186, 187, 188, 189, 190, 195, 196, 197,
 198, 199, 200, 201, 202, 203, 204, 205, 206, 211,
 212, 213, 214, 215, 216, 217, 218, 219, 220, 221,
 222, 227, 228, 229, 230, 231, 232, 233, 234, 235,
 236, 237, 238, 242, 243, 244, 245, 246, 247, 248,
 249, 250, 251, 252, 253, 254

A.3.3.5 IQ5 Huffman table generation parameters.

dc_luminance_bits[16] = 0, 3, 1, 1, 1, 1, 0, 0, 6, 3, 0, 0, 0, 0, 0, 0

dc_luminance_val[16] = 0, 2, 3, 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

ac_luminance_bits[16] = 0, 1, 3, 2, 5, 2, 3, 8, 1, 2, 1, 1, 1, 1, 0, 195

ac_luminance_val[226] = 1, 0, 2, 17, 33, 49, 3, 18, 65, 81,
 97, 113, 129, 34, 145, 161, 4, 19, 50, 177,
 193, 209, 225, 240, 66, 35, 82, 98, 241, 20,
 51, 114, 130, 146, 67, 162, 194, 178, 210, 5,
 6, 7, 8, 9, 10, 11, 12, 13, 14, 21,
 22, 23, 24, 25, 26, 27, 28, 29, 30, 36,
 37, 38, 39, 40, 41, 42, 43, 44, 45, 46,
 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
 62, 68, 69, 70, 71, 72, 73, 74, 75, 76,
 77, 78, 83, 84, 85, 86, 87, 88, 89, 90,
 91, 92, 93, 94, 99, 100, 101, 102, 103, 104,
 105, 106, 107, 108, 109, 110, 115, 116, 117, 118,
 119, 120, 121, 122, 123, 124, 125, 126, 131, 132,
 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
 147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
 157, 158, 163, 164, 165, 166, 167, 168, 169, 170,
 171, 172, 173, 174, 179, 180, 181, 182, 183, 184,
 185, 186, 187, 188, 189, 190, 195, 196, 197, 198,
 199, 200, 201, 202, 203, 204, 205, 206, 211, 212,
 213, 214, 215, 216, 217, 218, 219, 220, 221, 222,
 226, 227, 228, 229, 230, 231, 232, 233, 234, 235,
 236, 237, 238, 242, 243, 244, 245, 246, 247, 248,
 249, 250, 251, 252, 253, 254